

A New Estimate of the Hubble Time with Improved Modeling of Gravitational Lenses

Jonathan Coles

ABSTRACT

This paper examines free-form modeling of gravitational lenses using Bayesian ensembles of pixelated mass maps. The priors and algorithms from previous work are clarified and significant technical improvements are made. Lens reconstruction and Hubble Time recovery are tested using mock data from simple analytic models and recent galaxy-formation simulations. Finally, using published data, the Hubble Time is inferred through the simultaneous reconstruction of eleven time-delay lenses. The result is $H_0^{-1} = 13.7^{+1.8}_{-1.0}$ Gyr ($H_0 = 71^{+6}_{-8}$ km s⁻¹ Mpc⁻¹).

Subject headings: gravitational lensing; cosmological parameters

1. Introduction

Gravitational lenses provide a fantastic natural tool for probing many of the large scale properties of the cosmos. Recent applications range from estimating the age of the Universe (Saha et al. 2006) to studying the dark matter profiles of galaxies (Read et al. 2007) to testing alternative theories of gravity (Ferreras et al. 2007).

Despite their potential, gravitational lenses (GLs) are difficult to study because of several degeneracies such as the position of the source and the mass distribution of the lensing object. This paper focuses on strong lensing of quasars by galaxies, but the techniques developed can equally be applied to clusters. Many have tried to fit models to GLs by assuming different galaxy structures. Young et al. (1981) were the first to do so with King models and many others have followed using a variety of single isothermal ellipses (SIEs), Sèrsic models, or de Vaucouleurs profiles (for a review see Kochanek 2004). But different models can easily give different results (Vuissoz et al. 2007).

This kind of modeling is generally called parametric modeling. Each model has a nominal amount of parameters that can be adjusted. But while one model may fit the data, the degeneracies make it difficult to determine how well these models really represent the lens; and as pointed out by Bernstein & Fischer (1999) and more recently by Read et al. (2007),

in connection with time delays, without extreme care these models can be very sensitive to the assumptions.

In contrast, free-form or non-parametric models reconstruct the lens on a grid or a set of basis functions. No particular form is assumed and the results allow a wider range of solutions than parametric models might. Such modeling is not unique to lensing, though.

Schwarzschild (1979) used non-parametric modeling to show for the first time that it is possible to construct a triaxial stellar system in equilibrium. He showed that there existed a distribution of stars on orbits that fit a given density function D . The three-dimensional space of a galaxy was divided into M cells and D was expressed by $D(J) = \sum_{I=1}^M C(I) \cdot B(I, J)$, where $B(I, J)$ is the orbit density for an orbit I in cell J , calculated using test particles in a fixed potential. $C(I)$, the number of stars on orbit I , was determined numerically by solving a linear program.

In a very similar manner, Schwarzschild’s technique can be applied to lenses. Modeling the lenses on a grid was first introduced by Saha & Williams (1997) and then later extended to include both weak and strong lensing by AbdelSalam et al. (1998). Similar methods have also been used by Diego et al. (2005) and Bradač et al. (2005). But in contrast to Schwarzschild, it is desirable to show the *variety* of solutions rather than just existence. This important feature is incorporated into the work of Williams & Saha (2000) and the software `PIXELENS` (Saha & Williams 2004) (see Appendix A). Related approaches are developed in Trotter et al. (2000) and Keeton & Winn (2003). Given a large ensemble of models, one or several variables are examined while averaging out (marginalizing) the others. The same principle is used in statistical mechanics. However, the use of marginalization is sometimes overlooked, leading to a misunderstanding that pixelated models are “grossly underconstrained” because the number of variables exceeds the number of data points.

Pixelated modeling has the advantage of allowing the form of the lens to vary. It does not presuppose important parameters and can produce models that would otherwise not be possible with parametric modeling. For instance, while parametric models already showed that steepness is an important parameter (Wambsganss & Paczynski 1994), pixelated models showed that shape degeneracies, which are often difficult to capture with parametric models, cannot be ignored (Saha & Williams 2006); twists and nonuniform stretching are also easily found.

In this paper, pixelated lens modeling and the constraints imposed on the models are explicitly defined. The algorithms are improved with several optimization techniques and the enhanced method is tested against lenses from an N -body simulation and another fictitious data set. Finally, a system of eleven lenses is used in the same way as Saha et al. (2006) to

further constrain the Hubble Time.

2. Creating Models

PIXELENs generates an ensemble of lens models that fit the input data. In the Bayesian way, the ensemble itself provides estimates and uncertainties. Each model consists of a set of n discrete mass squares with density κ_n , a source position $\vec{\beta}$, and optionally, a variable h which is proportional to H_0 . If the time delays are unknown, the value of h is fixed. In this paper, where the time delays are known, h varies across the ensemble. The positions of observed images and the redshifts of the source and lens are taken to be given with errors small enough to be ignored. Time delays between images, when available, are similarly assumed to be accurate. Tests from Saha et al. (2006) show that adjusting these numbers slightly to simulate errors has much less effect than the model uncertainties.

The mass density in each square, or *pixel*, is the projected mass density on the plane of the sky in units of the critical density.¹ The pixelated surface is a disc of radius `pixrad` pixels. The total number of pixels is then about $\pi \cdot \text{pixrad}^2$. The extent of the modeled mass, `maprad`, defaults to $\min\{r_{\max} + r_{\min}, 2r_{\max} - r_{\min}\}$, where r_{\min} and r_{\max} are the distances of the innermost and outermost images, respectively. This allows for a buffer zone outside the outermost image when required.

Following Blandford & Narayan (1986), the arrival time is the light travel time scaled by

$$h^{-1}T(z_L, z_S) = (1 + z_L) \frac{D_L D_S}{c D_{LS}} \quad (1)$$

where z_L is the redshift of the lens, and D_L , D_S , and D_{LS} are the distances from observer to lens, observer to source, and lens to source, respectively. This removes the dependence on a particular cosmology. The h^{-1} factor comes through the distance factors.

The arrival time at position $\vec{\theta}$ is given by

$$\tau(\vec{\theta}) = \frac{1}{2}|\vec{\theta}|^2 - \vec{\theta} \cdot \vec{\beta} - \int \ln |\vec{\theta} - \vec{\theta}'| \kappa(\vec{\theta}') d^2 \vec{\theta}'. \quad (2)$$

This can be interpreted as a surface, which is modeled with a summation over the pixel

¹Many have suggested that it would be better to discretize the potential, but the potential is not naturally discrete and doing so would require recovering the mass from Poisson's equation; guaranteeing that the mass remains positive is difficult and involves a double derivative which produces noisy results.

densities,

$$\begin{aligned}\tau(\vec{\theta}) &= \frac{1}{2}|\vec{\theta}|^2 - \vec{\theta} \cdot \vec{\beta} - \sum_n \kappa_n Q_n(\vec{\theta}) \\ &+ \gamma_1(\theta_x^2 - \theta_y^2) + 2\gamma_2\theta_x\theta_y.\end{aligned}\tag{3}$$

Two additional terms involving γ_1 and γ_2 are added to account for external shear from neighboring galaxies.

The function Q is the integral from (2) evaluated over a square pixel with side length a . Q is defined using the same notation as in Saha & Williams (1997): Let x, y be the Cartesian components of $\vec{\theta}$, $r^2 = x^2 + y^2$ and

$$\begin{aligned}\tilde{Q}_n(x, y) &= (2\pi)^{-1} [\\ &+ x^2 \arctan(y/x) \\ &+ y^2 \arctan(x/y) \\ &+ xy(\ln r^2 - 3)]\end{aligned}\tag{4}$$

Then

$$\begin{aligned}Q_n(x, y) &= \tilde{Q}_n(x_+, y_+) + \tilde{Q}_n(x_-, y_-) \\ &- \tilde{Q}_n(x_-, y_+) - \tilde{Q}_n(x_+, y_-),\end{aligned}\tag{5}$$

where $x_{\pm} = x - x_n \pm a/2$ and $y_{\pm} = y - y_n \pm a/2$.

The function τ is linear in all the unknowns $\vec{\beta}, \kappa_n, \gamma_1, \gamma_2$. Constraints are placed on τ and the unknowns so that the results are physical. The data constraints come directly from lensing theory. The priors, or assumptions, are additional constraints that are physically motivated.

As a side note, the source position can be negative because it is relative to the center, but it must be positive in order to encode it as part of the linear program. This is resolved by adding a constant internally.

Data Constraint 1 *Images are observed where the arrival time surface is stationary, $\vec{\nabla}\tau(\vec{\theta}_i) = 0$ (Fermat's Principle).*

$$\begin{aligned}\vec{\theta}_{i,x} - \vec{\beta}_x - \sum dQ/d\vec{\theta}_{i,x} &= 0, \\ \vec{\theta}_{i,y} - \vec{\beta}_y - \sum dQ/d\vec{\theta}_{i,y} &= 0,\end{aligned}\tag{6}$$

Data Constraint 2 *The time delay between two images $\vec{\theta}_i$ and $\vec{\theta}_j$ must be consistent with observations,*

$$\tau(\vec{\theta}_i) - \tau(\vec{\theta}_j) = h \frac{[\text{obs delay}]}{T(z_L, z_S)}.\tag{7}$$

If the time delays are unknown the time ordering can be inferred from the morphology and imposed by

$$\tau(\vec{\theta}_i) - \tau(\vec{\theta}_j) \geq 0. \quad (8)$$

Data Constraint 3 *At each θ_i there are two constraints of the form*

$$\epsilon \left| \frac{\partial^2}{\partial \theta_{x'}^2} \tau(\vec{\theta}_i) \right| \leq \left| \frac{\partial^2}{\partial \theta_{y'}^2} \tau(\vec{\theta}_i) \right| \quad (9)$$

where $\theta_{x'}$ and $\theta_{y'}$ are the local radial and tangential directions and $\epsilon = 1/10$ by default.

This ensures that the image elongation is between ϵ and $1/\epsilon$ when projected along the radial direction. In practice, the default does not place any constraints on the image. If an image is known to be elongated then ϵ can be changed. In particular, this was used in AbdelSalam et al. (1998).

Data Constraint 4 *If a model contains N lenses, they must share the same Hubble Constant.*

$$h_{\text{lens}_1} = h_{\text{lens}_2} = \dots = h_{\text{lens}_N} \quad (10)$$

When H_0 is unspecified then H_0 is allowed to vary from model to model but not from lens to lens within a single model.

The following priors are the assumptions made about the lensing systems. All are well-defined and astro-physically justified, as explained below.

Prior 1 *The density cannot be negative.*

$$\kappa_n \geq 0 \quad (11)$$

This is a quite trivial requirement, but one that can often be difficult to ensure with other techniques. The linear programming algorithm employed here guarantees this prior by design.

Notice the similarity between Schwarzschild’s equation from the introduction on the one hand and Equation 3 and Prior 1. There is a linear function (D or τ) whose value is known, and a summation over a product where one of the product terms is calculated beforehand (B

or Q). Schwarzschild was limited at the time to what he could say about the unknowns, but negative values were not allowed. The goal was only to show the existence of one solution because no one knew at the time whether a triaxial solution was possible. With lenses much more can be said about the unknowns and lensing is known to occur.

Prior 2 *Most lens are assumed to have inversion symmetry, unless the lenses are observed to be interacting or otherwise strongly asymmetric.*

$$\kappa_{i,j} = \kappa_{-i,-j}. \quad (12)$$

Prior 3 *The density gradient should point within $\theta = 45^\circ$ of the center.*

$$\begin{aligned} \begin{bmatrix} i & j \end{bmatrix} M \nabla \kappa_{i,j} &\geq 0, \\ \begin{bmatrix} i & j \end{bmatrix} M^T \nabla \kappa_{i,j} &\geq 0, \end{aligned} \quad (13)$$

where

$$M = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (14)$$

$$\nabla \kappa_{i,j} \equiv (2a)^{-1} (\kappa_{i+1,j} - \kappa_{i-1,j} - \kappa_{i,j+1} - \kappa_{i,j-1}) \quad (15)$$

and a is the pixel size.

This complicated expression is just saying that if the density gradient of a pixel were pointing at most θ away from the center then moving the pixel's position by θ should align the density gradient so that it points directly at the center. If the gradient is greater than θ the “ \geq ” condition will not be satisfied.

Prior 4 *The density of a pixel must be no more than twice the average density of its neighbors.*

$$\kappa_n \leq 2 \frac{1}{N(n)} \sum_{i \in N(n)} \kappa_i, \quad n \neq 1 \quad (16)$$

This is a weak smoothing criterion. Normally, it is not applied to the central pixel, which can have arbitrary density.

Prior 5 *The mass profile must be steeper than r^{-s} . Let R_i be the set of all pixels on a discretized “ring” i of radius r_{R_i} , one pixel thick. The number of pixels in a ring is $|R_i|$. Let $C_i = r_{R_i}^s / |R_i|$, then*

$$C_i \sum_{n \in R_i} \kappa_n - C_{i+1} \sum_{n \in R_{i+1}} \kappa_n \geq 0. \quad (17)$$

The default radial mass profile constraint has $s = 0.5$. This is intentionally rather shallow, but as explained in Saha & Williams (2004) this is motivated by evidence showing that total density distribution in central regions of ellipticals is roughly isothermal, i.e. r^{-2} . Furthermore, the projected gas density in the Milky Way scales as $r^{-1.75}$ (Binney et al. 1991).

Again, the most important thing to realize from the constraints and the discretized lens equation is that *the constraints are all linear*. They can therefore be solved using any number of linear programming techniques. However, rather than find one solution, the space of all solutions is sampled to understand the distribution.

2.1. Bayesian MCMC Sampling

The linear equations presented in Section 2 constrain the solution space to a convex multi-dimensional polyhedron known as a simplex. The interior points of the simplex are solutions to the linear equations.

PIXELENs samples the interior points using a Monte-Carlo Markov-Chain (MCMC) technique. The general technique is described in condensed matter texts (Binney et al. 1992) and Bayesian books (Saha 2003). Each solution is used to reconstruct the arrival time surface, mass density contours, H_0^{-1} , etc.

The sampling method, Algorithm S, relies on being able to find random vertices of the simplex. The current implementation uses the standard linear programming Simplex Algorithm (Dantzig 1963; Press et al. 1986; Cormen et al. 2001) to maximize a given *objective function* subject to the linear constraints that form the simplex. The maximum is guaranteed to be at a vertex. For the present purposes, the objective function is chosen randomly after each iteration of Algorithm S, thereby producing a new vertex each time.

Algorithm S (*Sample interior points*)

1. Let γ_0 be a vertex on the simplex and $i = 0$ the index of the current iteration.
2. Let α_i be a new vertex.
3. Extend a line from α_i through γ_i until a constraint is reached. Select an interior point γ_{i+1} uniformly from the line.
4. If another model is desired, increment i and go back to 2, otherwise stop.

Because the simplex is convex by construction of linear hyperplanes, Algorithm S is guaranteed to return models in the solution space.

In addition to the explicit priors of the previous section, there is also a prior imposed by the sampling strategy itself. Although clearly well-defined, the physical significance of this prior continues to be the subject of study. This is not a point to be lightly dismissed since it influences the derived distribution of H_0 . However, the strategy cannot be arbitrary and there are very strict requirements on the way the volume can be sampled, which are discussed below. Numerous tests, both in this paper and others, have shown that the weighting can be empirically justified. The key point is that many different models must be examined. Other modeling techniques tend to assume the correctness of the model that is fit to the data, rather than letting the data itself reveal the model. To quote Blandford & Kundic (1997): “We should still aggressively explore all other classes of models that can also fit the observations but yet which produce disjoint estimates for the time delay. The true uncertainty in the Hubble constant is given by the union of all of these models.”

Algorithm S, in effect, puts a metric on the union. Previous PIXELENS papers implied that the sampling of the simplex was uniform in volume, but this is not correct, nor is it desired. The space does not have a Euclidean metric and while it is still unclear what metric the space *should* have, there are certain properties that an algorithm sampling the space *must* have.

1. The sampling strategy must be insensitive to changes in dimensionality of the space. In other words, increasing the number of variables (e.g. by increasing the pixel resolution, which subdivides pixels) should not change the predicted values of H_0 . This is not true if the solution space is uniformly sampled. As an example of the problem, imagine a uniformly sampled right triangle where the legs meet at the origin. The density of points projected onto one axis will be greater towards the origin. In higher dimensions, when the points are projected onto the same axis, the density distribution will be skewed further towards the origin.
2. The sampling strategy must be insensitive to units. The variables that define the solution space do not all have the same units. Some are mass density, some are source positions, one is H_0 , etc. By simply scaling any of these units the space is stretched or compressed. This would affect a sampling strategy based on volume when the number of dimensions is greater than two. Whatever the sampling prior is, it *must* be insensitive to this.

Both of these serious problems are solved by Algorithm S. The first problem is solved because a point is chosen uniformly along the line in step 3 regardless of the number of

dimensions. One can see from Figure 1 that the predicted value of H_0 remains quite steady even as the number of variables is increasing. The second problem is solved because the vertices of the space are used to guide the sampling strategy. How the vertices are chosen is completely independent of units. Any scaling would not affect the vertex selection procedure. Figure 2 shows a three dimensional sampled simplex. The sampling is clearly not uniform but is insensitive to stretching of the axes.

Algorithm S has changed slightly from older versions of `PIXELEN`S. Previous versions took a fixed number of vertex steps. The new vertex was often very close to the old one and resulted in clumps of correlated models. The new version seeks out vertices further away, which reduces the problem and better samples the interior with fewer samples. The running time increased with this change, but the results are more representative. Within the errors, though, old results are still valid.

Although Algorithm S does not sample the volume uniformly, in the limit of infinite samples, it does have *some* distribution. But how well is that distribution recovered with only a finite number of samples? To approximate the true distribution ten thousand models of the lens B1115+080 were generated. The “finite” sample consisted of 200 models. Figure 3 compares the distribution of just the Hubble Time variable. When the two samples are taken from the same distribution, the crosses fall on the dashed line. Even with a small sample, the distribution is well recovered.

2.2. Technical Issues

While `PIXELEN`S is stable, variations on sampling can introduce numerical instability. If a point is not chosen uniformly from the line in step 3 of Algorithm S a numerical error in the coordinates of sampled points will grow exponentially fast and lead to future points lying *outside* the solution space. Reprojecting a point back into the space is impractical because the exact size and shape of the simplex is unknown and truly incalculable due to the extraordinarily large number of dimensions and vertices. (It is worth noting that if all the vertices could be known in advance, the Simplex Algorithm would be unnecessary. One could simply pick a new vertex from the list.) In the worst case, however, this error is detectable. If such an error is detected the program will issue a message and halt.

The source of the error can be seen in Figure 4. The figure has been exaggerated for clarity. Sample points are constrained to lie on the shaded surface. After sampling points A and B , point C is the next intended point, but because of the limits of machine accuracy C' is taken instead.

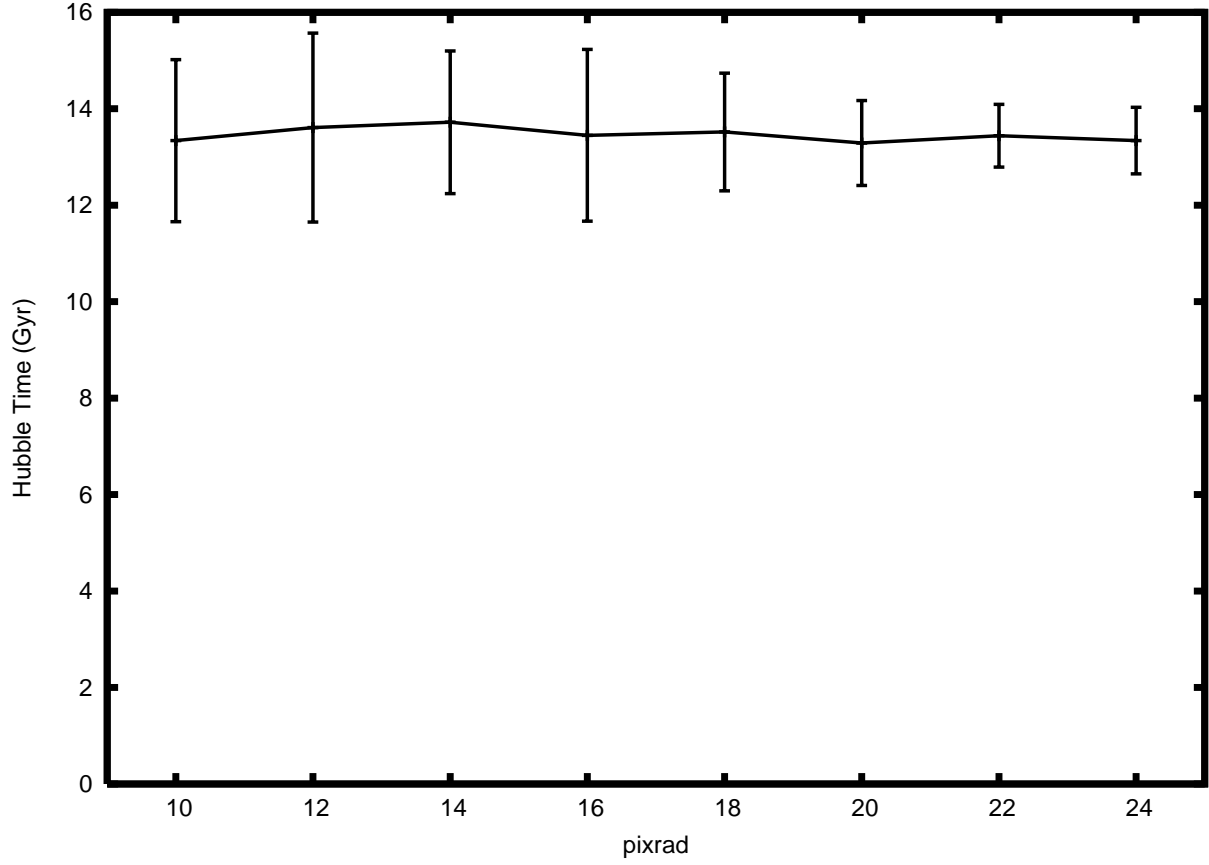


Fig. 1.— The predicted Hubble Time as a function of the pixel radius of the grid. The number of variables is $O(\text{pixrad}^2)$. Error bars indicate the 1σ deviations from the medians. Increasing the variables does not grossly affect the median H_0 , showing that condition (1) of the sampling strategy is satisfied. A single lens, B1115+080, is being modeled.

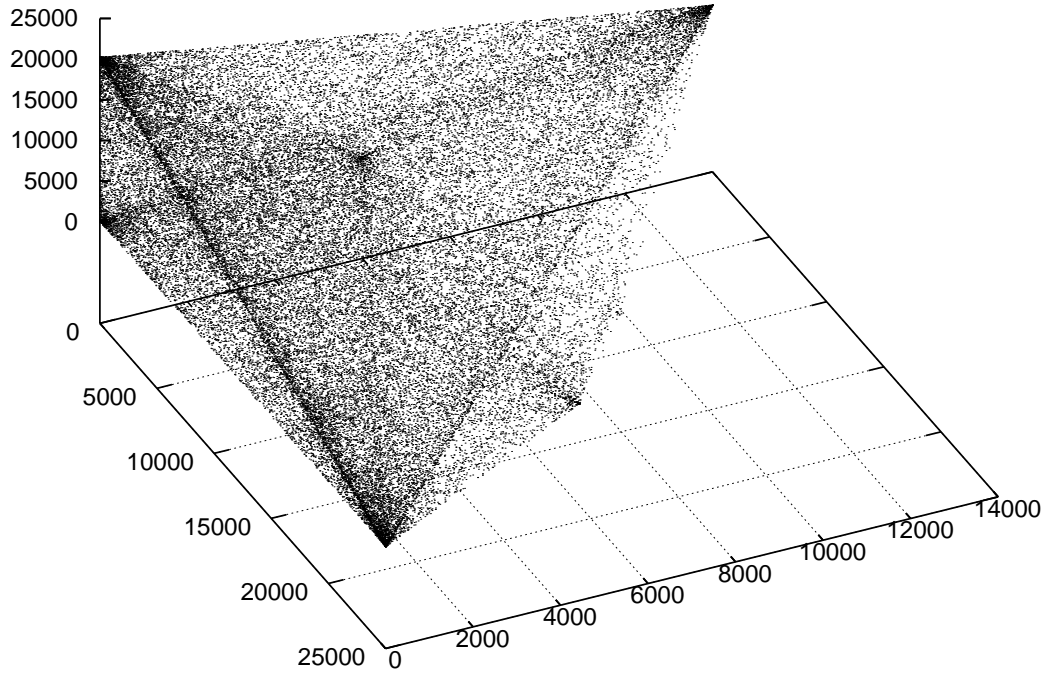


Fig. 2.— A three dimensional example of a sampled simplex with 50,000 points. The overdensities clearly indicate that the volume is not uniformly sampled. This must be the case in order to satisfy the two conditions of the sampling strategy.

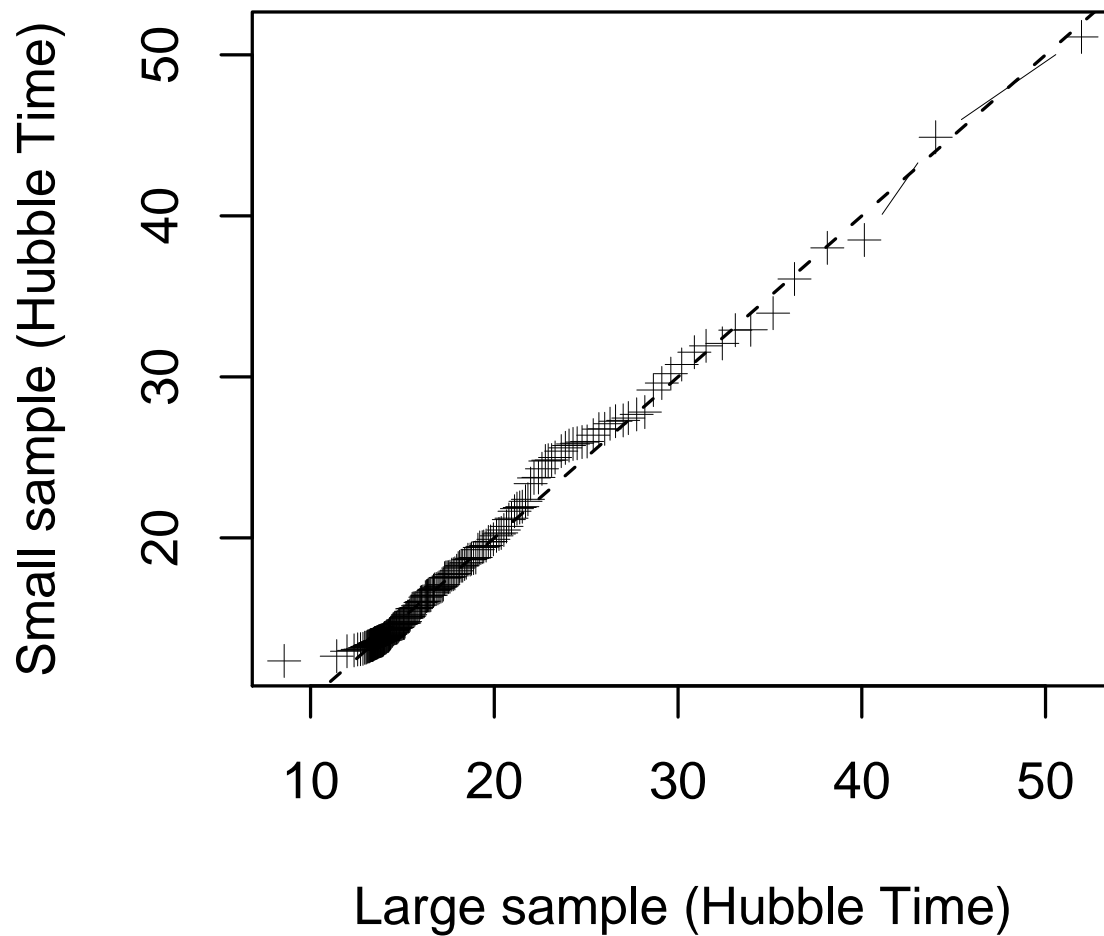


Fig. 3.— Quantile-quantile plot comparing the distribution of a large sample of Hubble Times to the distribution of a small sample. The points lie nearly perfectly on the dashed line, indicating that the two samples come from the same distribution. The tail extends off the figure because of a few extreme outliers in the large sample. The figure was clipped for clarity.

If the problem only occurred once, the error would be below the noise in the system, but each sample introduces more error because the next sample depends on the position of the previous sample.

Using the notation of Algorithm S, the further γ_{i+1} is chosen from γ_i the larger the error. This is a simple lever; the error is proportional to (a/b) where $a = \gamma_{i+1} - \alpha_i$ and $b = \gamma_i - \alpha_i$. Successive errors are compounded over N iterations:

$$\epsilon = \prod_i^N (a_i/b_i). \quad (18)$$

Sampling uniformly along the line suppress the error because points are chosen close to γ_i as often as far away. If $a_i \geq b_i$, ϵ grows without bound. If $\ln \epsilon > 0$ then $\langle a/b \rangle \geq N$, in which case the error is reported and the program halts.

A number of technical improvements were also made to the implementation of the Simplex Algorithm. As mentioned earlier, the Simplex Algorithm is used to find a new vertex in Algorithm S by maximizing an *objective function* subject to the linear constraints that form the simplex. Each iteration moves to a new vertex that increases the objective function until no further vertex can be found. The linear constraints are stored in a matrix called a *tableau*. The algorithm moves to the next vertex by rewriting the tableau; an operation known as a pivot. For very large problems the pivot is the bottleneck. This work improves the performance by parallelizing the pivot on a shared memory machine. For even larger problems than are faced here it may be necessary to extend this to a distributed-memory cluster of machines.

A further improvement was an optimization of the data structure used to store the tableau. While the tableau is initially sparse, and previous versions of PIXELENs stored it as such, the tableau quickly becomes dense after only a few pivots (Figure 5). Storing the tableau as a dense matrix yields a significant performance boost.

3. Testing Hubble Time Recovery

How well does Algorithm S predict the Hubble Time? Two tests were performed.

First, a blind test similar to that in Williams & Saha (2000). Four quad lenses were crafted assuming a particular Hubble Time that was unknown to the author. These were, in fact, the same lenses as in the aforementioned paper, but rescaled to a Hubble Time of 13.9 Gyr. The time delays were perturbed slightly to simulate errors. The Hubble Time was recovered using PIXELENs and then the simulated Hubble Time revealed. Figure 6

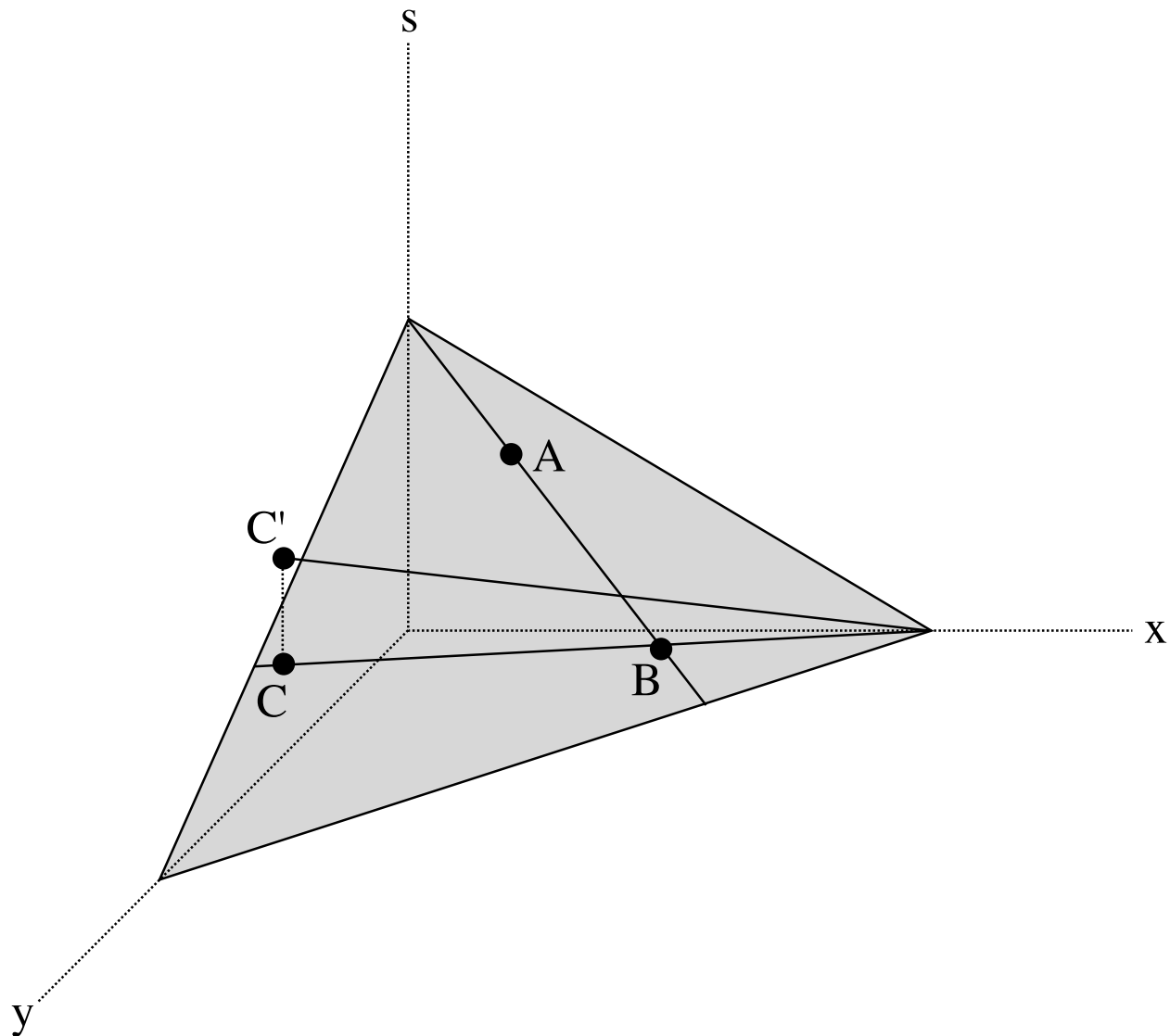


Fig. 4.— Example of numerical error in selecting point C . The x and y axes are the two main variables, and s is the slack variable introduction by the Simplex linear programming algorithm. The grey region is the plane on which solutions lie. Point C lies far enough from point B that numerical error is introduced, leading to the selection of C' , which lies outside the grey solution space. Subsequent similar sampling leads to exponentially fast growing error. The error in the diagram is exaggerated for clarity.

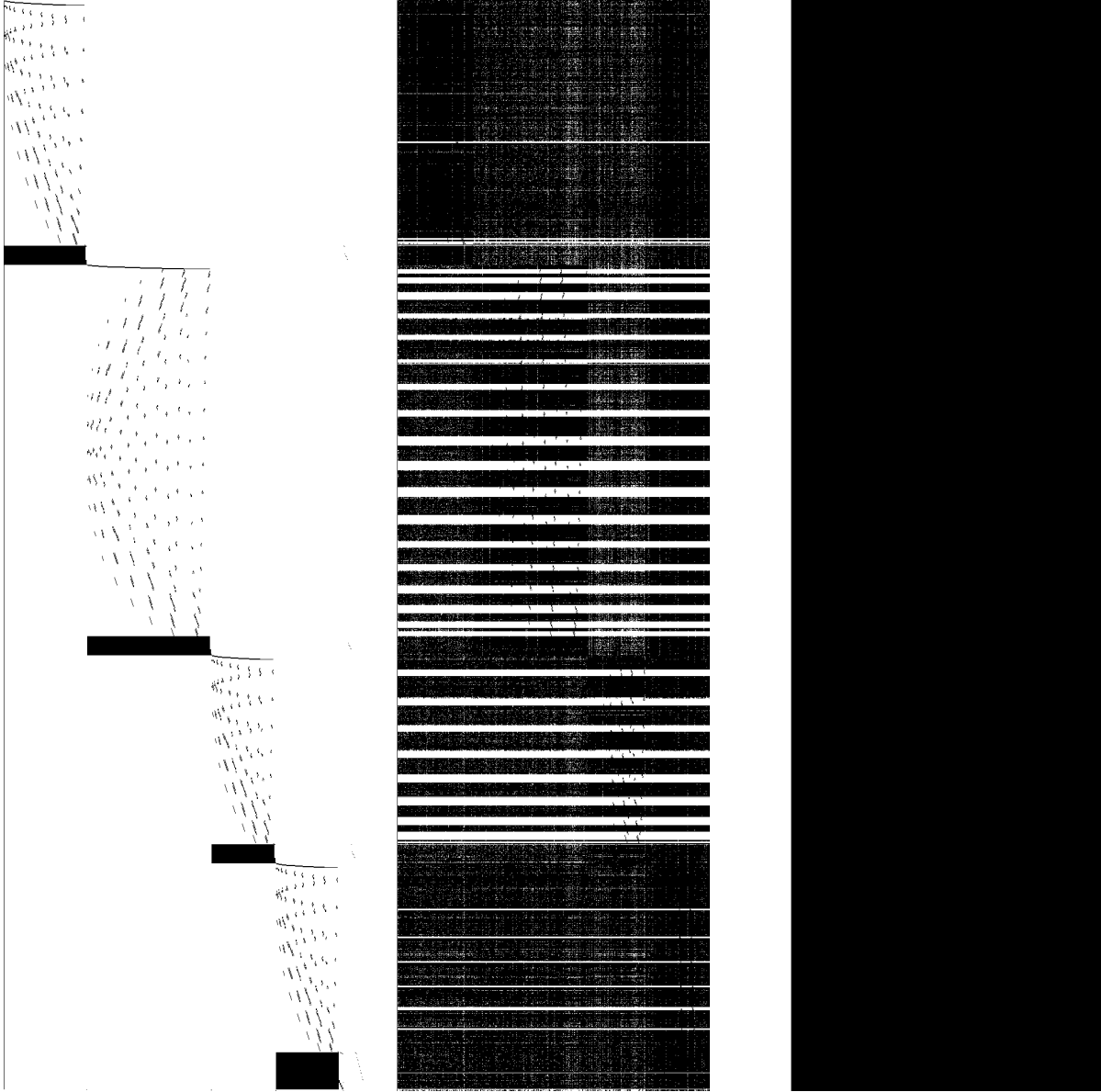


Fig. 5.— Three simplex tableaux displayed graphically. The left image is the tableau with the original constraints in place. Lens asymmetry can be seen in the block that is twice as tall as the other three. The middle image is after a feasible solution is found. The third image is after 200 models. Black represents non-zero values.

shows the histogram of Hubble Times from two hundred models. **PIXELEN**s predicts $H_0^{-1} = 13.7_{-1.4}^{+1.5}$ Gyr.

Second, five lenses, three doubles and two quads, were created by ray-tracing a galaxy from the N -body plus hydrodynamic simulation with $H_0^{-1} = 14$ Gyr described by Macciò et al. (2006). The galaxy is an E1 or E2 triaxial elliptical with about 80% dark matter. The histogram of Hubble Times from two hundred models is shown on the right in Figure 6. There is a clear peak with the predicted value at $H_0^{-1} = 13.3_{-0.6}^{+1.4}$ Gyr with 68% confidence. Within the errors **PIXELEN**s successfully recovers the simulation Hubble Time. Read et al. (2007) reconstruct the same lenses with a slightly different prior.

4. New 11-Lens Results

With confidence founded in the results of the last section, an ensemble of lenses was modeled to find the true Hubble Time. Saha et al. (2006) used ten lenses² to constrain the Hubble Time to $H_0^{-1} = 13.5_{-1.3}^{+2.5}$ Gyr. Subsequently, Vuissoz et al. (2007) have reported on a new time delay measurements for J1650+4251. Combining this new lens measurement with the ten lenses used previous, all eleven lens were simultaneously modeled to predict tighter bounds on the Hubble Time. The distribution of Hubble Times is shown in Figure 7. At 68% confidence, the new predicted value is

$$H_0^{-1} = 13.7_{-1.0}^{+1.8} \text{ Gyr } (H_0 = 71_{-8}^{+6} \text{ km s}^{-1} \text{ Mpc}^{-1}).$$

Figure 8 shows the ensemble average of the mass and arrival time surface for J1650+4251 as recovered by **PIXELEN**s. Average mass maps for the other lenses are similar to those in Saha et al. (2006), Figure 2.

To put this into context, the results of other techniques are listed below. The units are in H_0 , which is found more often in the literature than H_0^{-1} . The latter appears more naturally in lensing, though, hence the presentation of the above estimates. The first set of errors are statistical and the second set (when applicable) are systematic. This list is summarized by the plot in Figure 9.

1. $H_0 = 73 \pm 3 \text{ km s}^{-1} \text{ Mpc}^{-1}$ from the cosmic microwave background fluctuation spectrum (Spergel et al. 2007). The Hubble Constant is just one value in a multiparameter fit.

²The ten lenses are J0911+055, B1608+656, B1115+080, B0957+561, B1104-181, B1520+530, B2149-274, B1600+434, J0951+263, and B0218+357.

2. $H_0 = 68 \pm 6 \pm 8 \text{ km s}^{-1} \text{ Mpc}^{-1}$ using a different Monte Carlo method to combine lenses (Oguri 2007).
3. $H_0 = 62.3 \pm 1.3 \pm 5.0$ (Sandage et al. 2006) and $H_0 = 73 \pm 4 \pm 5$ (Riess et al. 2005) from Cepheid-calibrated luminosity of Type Ia supernovae. This is independent of the global geometry.
4. $H_0 = 66_{-10-8}^{+11+9} \text{ km s}^{-1} \text{ Mpc}^{-1}$ from the Sunyaev-Zel’dovich effect (Jones et al. 2005). As with lensing, a global geometry is assumed and the Hubble Time is measured.
5. $H_0 = 72 \pm 8$ (Freedman et al. 2001) using a variety of Cepheid-calibrated indicators. This is again, independent of the global geometry.

In the future, better predictions may be obtained with improved priors and tighter constraints on galaxy structure. Simply adding more lenses will also improve the predictive power of `PIXELEN`S, but may not help in understanding the different sources of degeneracies and developing better priors.

5. Summary

Pixelated lens reconstruction is an example of free-form modeling. Such modeling has the advantage that one does not have to presuppose what the important parameters might be and can let the generated models be a guide to finding those parameters. Free-form modeling has many applications and was used early by Schwarzschild to show the existence of triaxial stellar systems in equilibrium.

Applied to gravitational lensing, the free-form models are implemented as pixelated models whereby the mass sheet of the lens is discretized into many small square pixels. The mass in each pixel is recovered using an MCMC technique using linear programming to probe the solutions which reconstruct the observed data. The software `PIXELEN`S produces an ensemble of hundreds of such models. The ensemble provides Bayesian statistics about the variety of possible lens reconstructions.

The constraints that define the mass models are explained. The linear constraints form a hyper-dimensional solution space from which the models are drawn. The sampling algorithm has been improved over previous software versions and although it was shown that the algorithm does not uniformly sample the solution space, it is argued that this is undesirable for this problem. The implementation was parallelized for multi-processor,

shared memory machines. Future work will include controlling numerical round-off errors that will become significant with larger problems.

The new version of `PIXELEN`S was applied to an ensemble of eleven lenses to determine a new value for the Hubble Time: $H_0^{-1} = 13.7_{-1.0}^{+1.8}$ Gyr within 68% confidence.

Further research into galaxy and cluster structure is needed to improve the priors. The estimates of galaxy morphology have been conservative but tighter constraints will lead to better results. Furthermore, model ensemble building can be applied to other areas, even to the original problems of Schwarzschild.

Pixelated lens modeling is on the cutting edge of gravitational lens research, promising to provide great insight into the structure of galaxies, the distribution of dark matter, and the fundamental nature of the Universe. But there are still many challenges both scientifically and computationally.

6. Acknowledgments

I would like to extend my sincere appreciation for the help I received with many aspects of this paper. In particular, Joachim Stadel for a critical insight concerning the material in Section 2.2; Peter Englmaier, Tristen Hayfield, and Justin Read for the hours spent considering different sampling techniques; and Prasenjit Saha for patiently answering my many lensing questions and ever so subtly nudging me to finish. I would also like to thank the anonymous referee for useful comments and suggestions on making the paper clearer and more concise.

REFERENCES

- AbdelSalam, H. M., Saha, P., & Williams, L. L. R. 1998, *AJ*, 116, 1541
- Bernstein, G. & Fischer, P. 1999, *AJ*, 118, 14
- Binney, Dowrick, Fisher, & Newman. 1992, *The Theory of Critical Phenomena* (Oxford, UK: Oxford University Press)
- Binney, J., Gerhard, O. E., Stark, A. A., Bally, J., & Uchida, K. I. 1991, *MNRAS*, 252, 210
- Blandford, R. & Narayan, R. 1986, *ApJ*, 310, 568

- Blandford, R. D. & Kundic, T. 1997, in *The Extragalactic Distance Scale*, ed. M. Livio, M. Donahue, & N. Panagia, 60–75
- Bradač, M., Schneider, P., Lombardi, M., & Erben, T. 2005, *A&A*, 437, 39
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. 2001, *Introduction to algorithms* (Cambridge, MA, USA: MIT Press)
- Dantzig, G. B. 1963, *Linear Programming and Extensions* (Princeton, NJ: Princeton University Press)
- Diego, J. M., Protopapas, P., Sandvik, H. B., & Tegmark, M. 2005, *MNRAS*, 360, 477
- Ferreras, I., Sakellariadou, M., & Furqaan Yusaf, M. 2007, *ArXiv e-prints*, 709
- Freedman, W. L., Madore, B. F., Gibson, B. K., Ferrarese, L., Kelson, D. D., Sakai, S., Mould, J. R., Kennicutt, Jr., R. C., Ford, H. C., Graham, J. A., Huchra, J. P., Hughes, S. M. G., Illingworth, G. D., Macri, L. M., & Stetson, P. B. 2001, *ApJ*, 553, 47
- Jones, M. E., Edge, A. C., Grainge, K., Grainger, W. F., Kneissl, R., Pooley, G. G., Saunders, R., Miyoshi, S. J., Tsuruta, T., Yamashita, K., Tawara, Y., Furuzawa, A., Harada, A., & Hatsukade, I. 2005, *MNRAS*, 357, 518
- Keeton, C. R. & Winn, J. N. 2003, *ApJ*, 590, 39
- Kochanek, C. S. 2004, *ArXiv Astrophysics e-prints*
- Macciò, A. V., Moore, B., Stadel, J., & Diemand, J. 2006, *MNRAS*, 366, 1529
- Oguri, M. 2007, *ApJ*, 660, 1
- Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. 1986, *Numerical Recipes: The Art of Scientific Computing*, 1st edn. (Cambridge (UK) and New York: Cambridge University Press)
- Read, J. I., Saha, P., & Maccio, A. V. 2007, *ArXiv e-prints*, 704
- Riess, A. G., Li, W., Stetson, P. B., Filippenko, A. V., Jha, S., Kirshner, R. P., Challis, P. M., Garnavich, P. M., & Chornock, R. 2005, *ApJ*, 627, 579
- Saha, P. 2003, *Principles of Data Analysis* (Great Malvern, UK: Cappella Archive)
- Saha, P., Coles, J., Macciò, A. V., & Williams, L. L. R. 2006, *ApJ*, 650, L17

- Saha, P. & Williams, L. L. R. 1997, MNRAS, 292, 148
- . 2004, AJ, 127, 2604
- . 2006, ApJ, 653, 936
- Sandage, A., Tammann, G. A., Saha, A., Reindl, B., Macchetto, F. D., & Panagia, N. 2006, ApJ, 653, 843
- Schwarzschild, M. 1979, ApJ, 232, 236
- Spergel, D. N., Bean, R., Doré, O., Nolta, M. R., Bennett, C. L., Dunkley, J., Hinshaw, G., Jarosik, N., Komatsu, E., Page, L., Peiris, H. V., Verde, L., Halpern, M., Hill, R. S., Kogut, A., Limon, M., Meyer, S. S., Odegard, N., Tucker, G. S., Weiland, J. L., Wollack, E., & Wright, E. L. 2007, ApJS, 170, 377
- Trotter, C. S., Winn, J. N., & Hewitt, J. N. 2000, ApJ, 535, 671
- Vuissoz, C., Courbin, F., Sluse, D., Meylan, G., Ibrahimov, M., Asfandiyarov, I., Stoops, E., Eigenbrod, A., Le Guillou, L., van Winckel, H., & Magain, P. 2007, A&A, 464, 845
- Wambsganss, J. & Paczynski, B. 1994, AJ, 108, 1156
- Williams, L. L. R. & Saha, P. 2000, AJ, 119, 439
- Young, P., Gunn, J. E., Oke, J. B., Westphal, J. A., & Kristian, J. 1981, ApJ, 244, 736

A. PixeLens Gravitational Lens Modeling Software

PIXELENs is freely available under the GNU General Purpose License. Source code is naturally included. The program is cross-platform and an I/O limited version even runs in a web browser. The version used in this paper is v1.88. For more information, visit <http://www.qgd.uzh.ch>.

Input data to PIXELENs used in this paper is available with the on-line version.

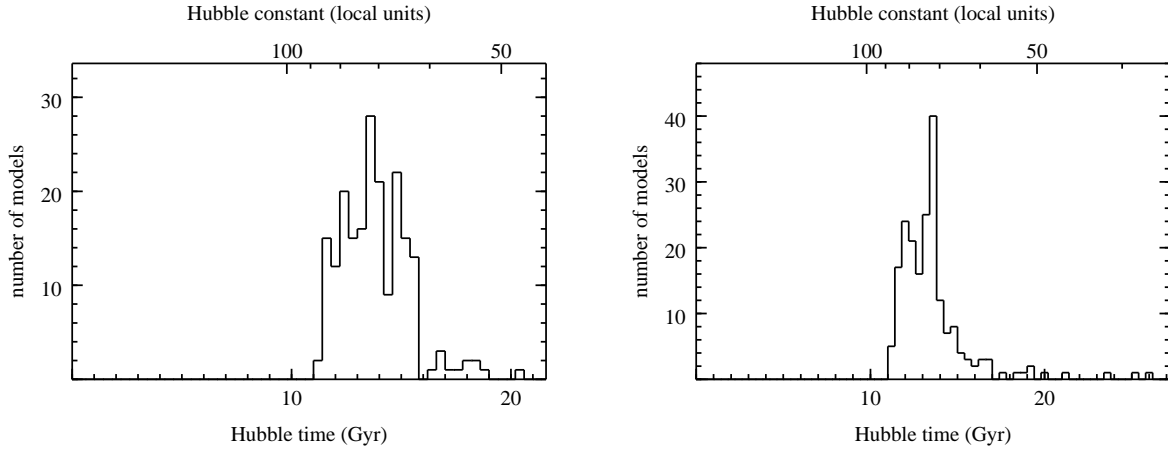


Fig. 6.— Two tests of the program. On the left are Hubble Time values recovered during a blind test. The lens was constructed by hand using an artificial value of the Hubble Time (13.9 Gyr). On the right, are time delays from a multiply lensed simulation galaxy with $H_0^{-1} = 14$ Gyr. There is a clear peak at 13.3 Gyr.

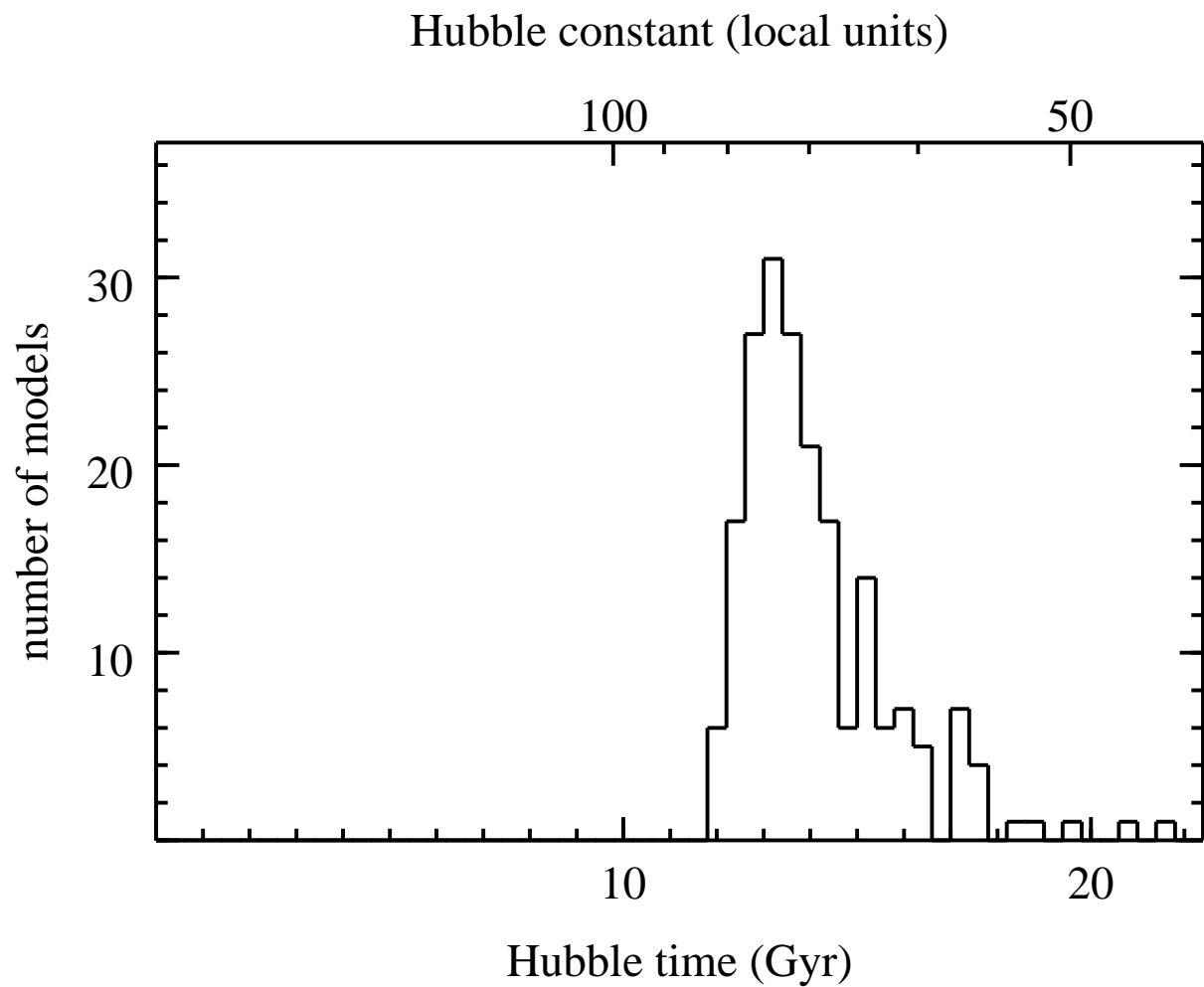


Fig. 7.— Hubble Time values found from simultaneously modeling eleven lenses. The peak occurs at 13.7 Gyr.

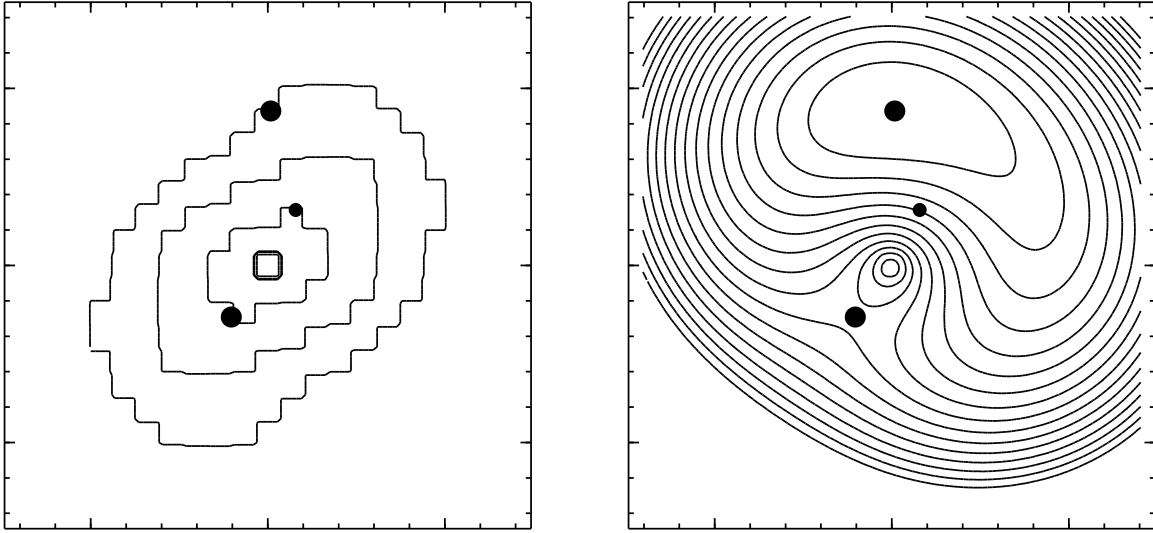


Fig. 8.— An example of the output from `PIXELEN`: The ensemble average of the mass (left) and arrival time surface (right) of J1650+4251.

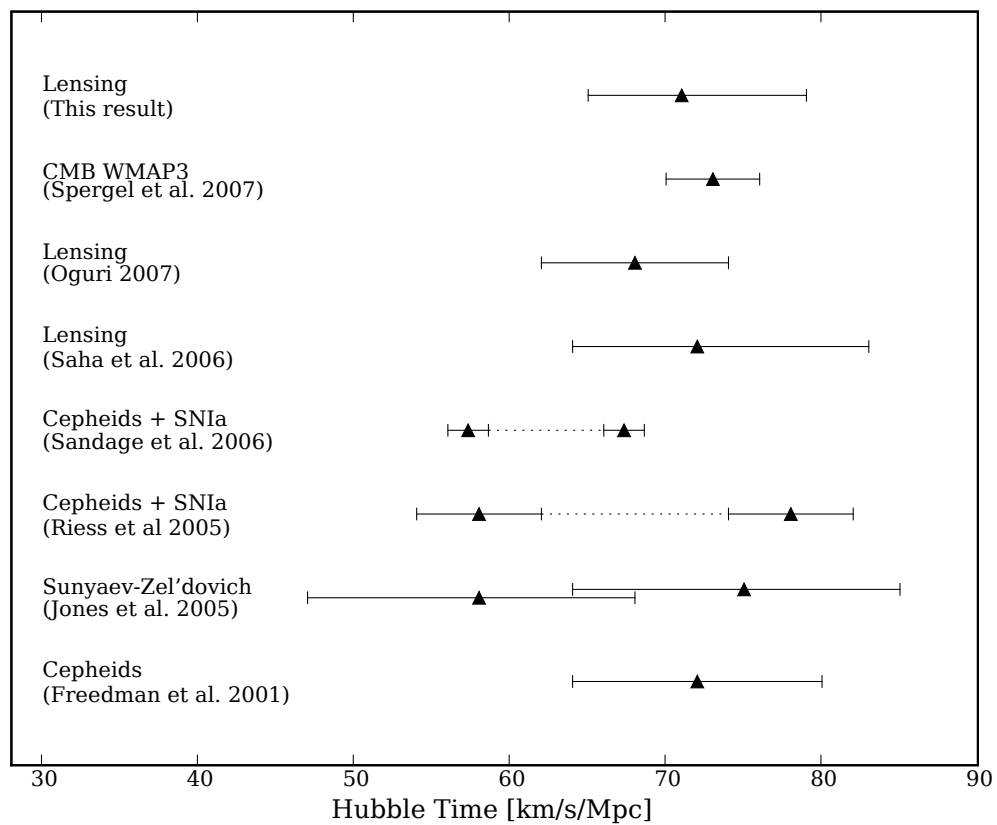


Fig. 9.— Recent Hubble Time measurements from a variety of methods. Multiple error bars for a single reference are present when there are systematic errors in addition to statistical.